An RTOS Based Architecture for Health Monitoring with Wireless Sensor Network

Sowmiya.P[#1], Valarmathy.G[*2]

[#]PG Student (CS), Assistant Professor (ECE) & G.K.M College of Engineering and Technology. Chennai, India.

[1]sowmiyaperumalsp@gmail.com

[2] gkmceteee001@gkmcet.net.in

Abstract— **An RTOS based architecture designed for the purpose of data transmission between two controlling units through WSN without collision. RTOS is used here to assign the priorities. The values are sent in the form of packets during specific time; in this transmission collision may occur to overcome this Wireless HART (Highway Addressable Remote Transducer) is used. This paper is implemented in two section as prototype. First section runs with RTOS and LPC2148 as master node and another as normal data acquisition node to which sensor are connected. Second section contains a controller. Communication between two nodes is accomplished through wireless HART. The basic view of this technique is to reduce the possibility of collision and thus increase the communication reliability. The sensing parameters can be displayed as graph in software Implementation. To do this, all the nodes must be synchronized precisely. Also, the stack designer must guarantee that the node can finish everything within the timeslot. It offers benefits in terms of product life cycle, safety, system integration complexity, performance scalability.**

Keywords— **RTOS, µC/OS-II, Wireless HART, Computing matches.**

I. INTRODUCTION

Wireless process control has been a popular topic recently in the field of health monitoring. Compared to traditional wired process control systems, their wireless counterparts have the potential to save costs and make installation easier. Also, wireless technologies open up the potential for new automation applications. Several industrial organizations, such as ISA[7], HART[3], WINA[8] and zigbee[10], have been actively pushing the application of wireless technologies in health monitoring is released, there have been a few publicly available standards on office and manufacturing automation, such as zigbee[10] and Bluetooth[2]. However, these technologies cannot meet the stringent requirements of industrial control. Compared with office applications, industrial applications have stricter timing requirement and higher security concern. For example, many monitoring applications are expected to retrieve updates from sensors every one second. Neither Zigbee nor Bluetooth makes any effort to provide a guarantee on end-to-end wireless communication delay. In addition, health environments are harsher for wireless applications in terms of interferences and obstacles than office environment. Some interference may be persistent. Zigbee, without built-in channel hopping technique, would surely fail in such environments. Bluetooth assumes quasi-static star network, which is not scalable enough to be used in large process control systems. The new RTOS based health monitoring is specifically targeted to solve these problems and provide a complete solution for process control applications. In this paper we discuss how we developed a prototype wireless sensor network protocol stack.

The Operating systems (OS) can be broadly divided into two classes (i) General Purpose Operating Systems (GPOS) and (ii) Real Time Operating Systems (RTOS). Members of the GPOS class use a ‘fair Scheduling’ algorithm to ensure that all tasks get a fair share of execution time; they do not take into account the priority of the task being executed. This is done to provide high system throughput, but can result in higher priority tasks getting delayed waiting for lower priority tasks to finish. The RTOs class, on the other hand, uses a priority based scheduling algorithm, where higher priority tasks are pre-emptively scheduled ahead of lower priority tasks, ensuring higher priority tasks are not starved of CPU time.

Not generating alarms in a timely manner can have catastrophic results for a patient. This makes the real-time performance of the local node a critical design parameter.

RTOS is a Process which is done between hardware and application. Each Monitoring system has their own priority among the various tasks according to their process. RTOS is used here to assign the priorities. Application code designed on an RTOS [11] can be quite diverse, ranging from a simple application for a digital stopwatch to a much more complex application for a aircraft navigation [10]. Here Micro C/OS II is used to assign the priorities [5]. µC/OS-II is a real-time pre-emptive multitasking embedded OS kernel. A pre-emptive kernel is used when system responsiveness is important; therefore, µC/OS-II is a completely portable, ROMable, scalable, real-time kernel. It provides services like task management, memory management and time management. µC/OS-II kernel supports pre-emptive algorithms such as round-robin, rate monotonic scheduling policy.

II. OVERVIEW ALGORITHM

Figure 1 shows high-level overview of the system. Operation can be decomposed into three phases. Data sampling: sample the sensors data of the structure, and logs it into processor memory. Data Collection: Transfer data reliably to an external computing resource. Data Analysis: runs analysis algorithm, and determines health status. Sends feedback to nodes if needed.

The system can also be decomposed into its structural pieces. This paper is structured according to the components of the system. The system can be decomposed into three pieces: hardware, software on the node, and analysis software on a central computer. Hardware issues including sensors and power will be presented in proposed design.

Fig 1 Overview of the system

Real-time patient monitoring conditions like heart failure, temperature exceeded are generated using sensors. Such sensors to be processed in real-time and requires significant signal processing with hard real-time limits. The real-time performance requirement of the monitoring system is a critical design parameter.

III. PROPOSED DESIGN ANALYSE

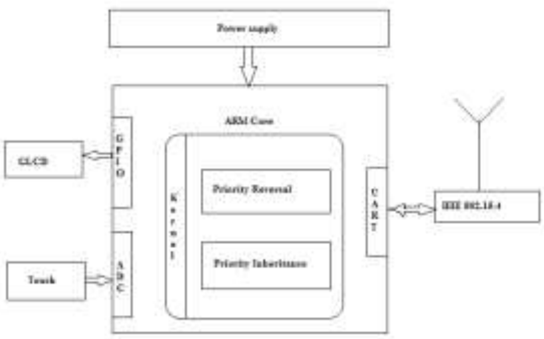

Fig 2 Master Node

Fig 2 shows the block diagram of the implemented master node system. It utilizes an RTOS running on an ARM controller to the GLCD, which displays the data in a browser window. Fig 3 shows the implemented data acquisition node of the system.

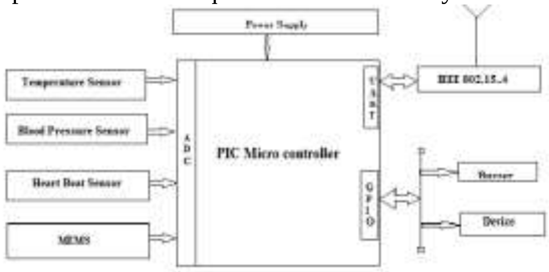

Fig 3 Data acquisition node

A. Sensors

Some sensors used in the system may send data simultaneously at rates ranging from 1Hz to greater than 1KHz. The aim here was to simulate high bit rate multi-sensor data being collected from an sensor array, e.g. multi-lead temperature and heart beat and blood pressure sensor etc. To measure the multi-sensor system performance five data streams were simulated, with each data stream transmitting a signal at 1 KHz. The five data streams may represent a combination of a 3-lead signal with two other data streams coming from other sensors or from other activity on the device. This simulated sensor data is then passed through an IEEE 802.15.4 device.

B. Master Node

Patient monitoring systems are designed to generate in the case of an emergency must be able to guarantee that they can do so immediately upon discovering a problem. GPOS are not up to this task, especially when under heavy processing load (as is often the case with devices such as personal computers, tablets and smart phones). Additionally, devices used for purposes other than remote monitoring activities run the risk of potentially significant downtime should some other process crashes the system.

µC/OS-II operating system is a real-time operating system that has been used for a variety of time-critical applications, and has been designed to meet extremely strict real-time requirements. The local node was implemented using the µC/OS-II. The hardware are used for the local node is the LPC 2148 controller. Because it is a comparatively low-cost device that meets the requirements for running the operating system. It utilizes an ARM 7, with 512MB of flash memory and a USD card reader that offers flexibility in short to medium-term data storage. In addition, it has built-in wireless an attached LCD touch screen display.

C. Data acquisition node

The data acquisition node designed using PIC 16F877A Controller. Here all the sensors are connected ADC port of PIC controller. This measured data's are transmitted using IEEE 802.15.4

network, and its enable the user module to watch the received data in real-time. In RTOS all the task execution time evaluation is important For example given task ‘Ti’ in OS is defined by:

- Arrival time Ai when task is on ready queue.
- The worst case execution time (WCET) Ci is the maximum time taken by system to complete given task.
- Start time Si is time when OS starts execution of the task.
- Finishing time Fi is time when OS completes the task.
- Di is hard real-time constraint.
- Di= Ai + DI

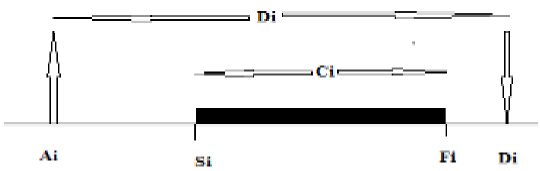

Fig 4 Parameters of Real-time task

Fig 4 shows the parameters of the real-time tasks. The time Di defines the time limit by which the task ‘Ti’ must be completed. This is the hard real-time limit. The execution time Ci is affected by other task and threads running on the system and must be managed by OS to ensure that Ci<Di.

Latency of the system is defined as =Si-Ai
Both the latency and execution time have significant impact if the OS is to meet is hard real time requirements.

Data Processing algorithms for most of the sensors use windowing for performing variety of tasks like filtering, spatial and frequency domain processing, segmentation, energy calculation, thresholding etc. All these tasks must be executed within timeframe for each window. This puts a hard real-time performance requirement on the mobile device. The execution time window for a given task decreases as sampling rate increases.

IV. EXPERIMENTAL RESULTS

A. Simulation Result

In this simulation three sensors will be connected to the Analog to Digital Converter. This will read the values from sensors. After Reading the values it will be given to the PIC Microcontroller. Microcontroller will accept only a 5 voltage so the sensor readings are converted as below 5v.

At end of the display unit the sensor values again converted into the normal values, then it will be displayed. The display value depend on the given switch condition in the switching.

The below figure 5 shows the design of the overall system. By changing the switching condition it can vary the Output of the Display unit. The Switching Operation depends on the low value and one high value for each sensor.

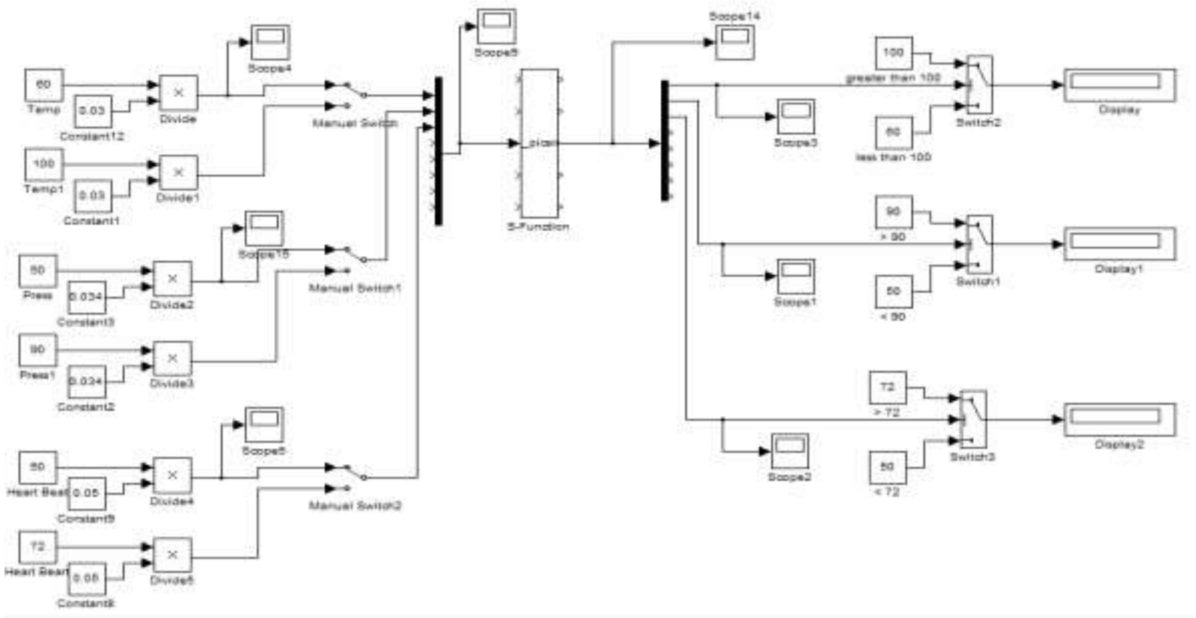

Fig 5 Full system Design

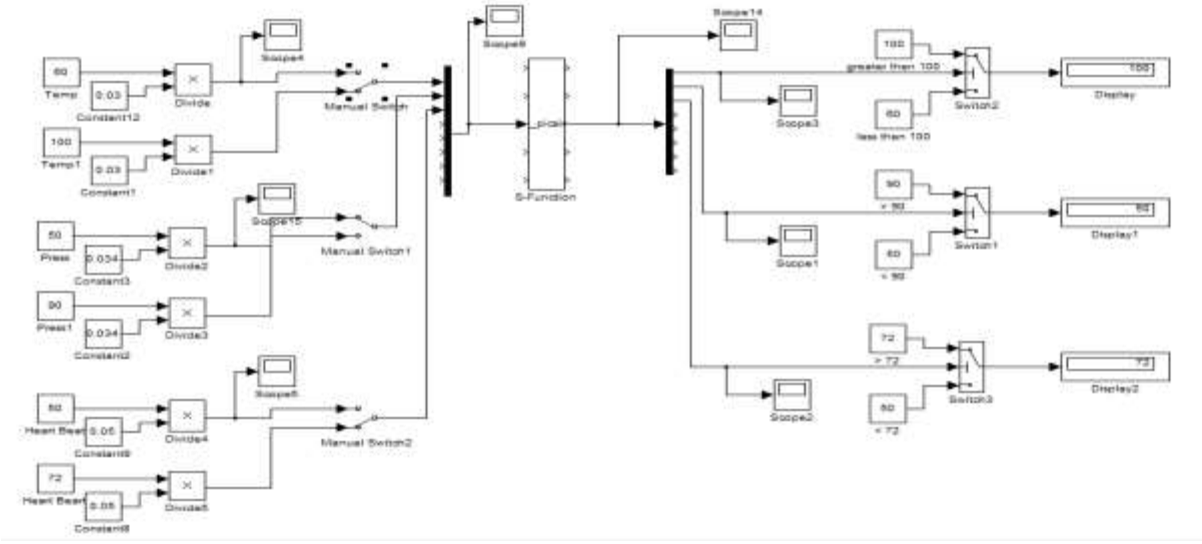

Fig 6 Temperature Sensor Output at 100

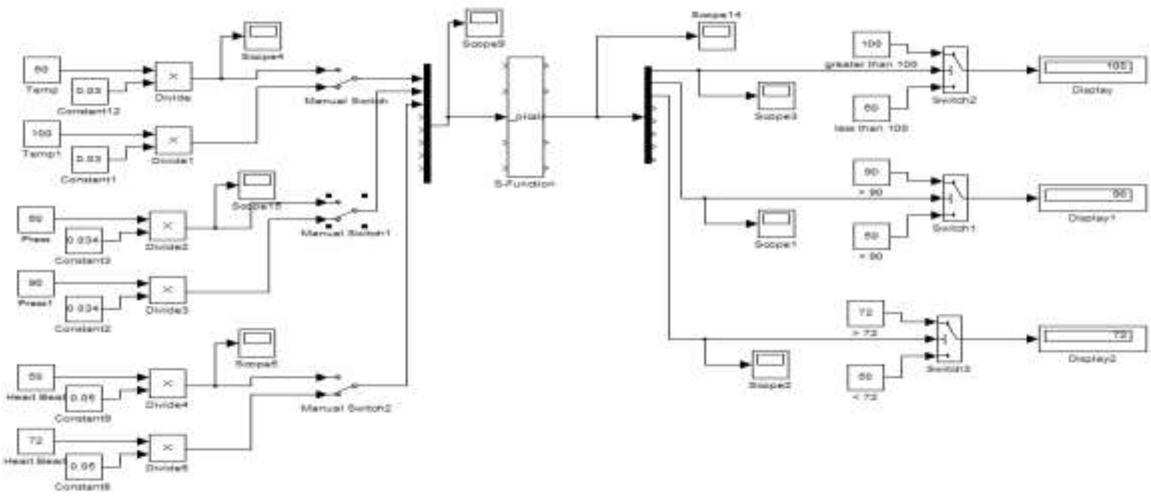

Fig 7 Pressure Sensor Output at 90

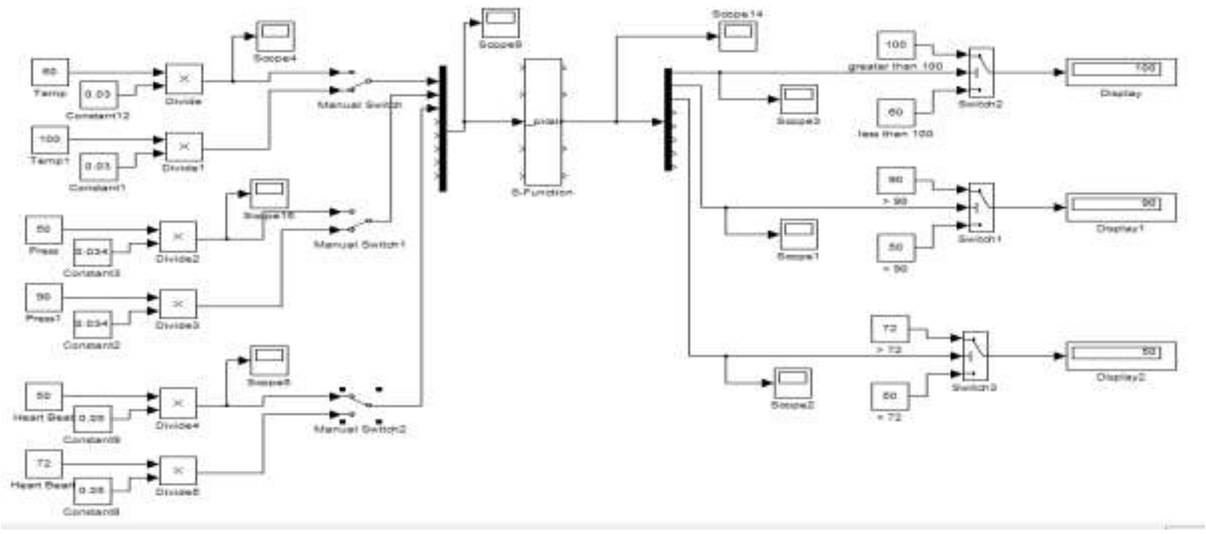

Fig 8 Heart beat sensor o/p at 50

B. Hardware Result

Fig 9 Hardware output

The Programming module is implemented using ARM. The system mainly consists of monitoring unit. Here PIC Microcontroller interfaced sensors used to measure the Patient Up normalities. This is connected to Zigbee through RS-232 cable. Then master node zigbee received the data and display the output in GLCD. In this master node priority scheduling process is performed.

V. CONCLUSION

Real-time Patient monitoring is becoming increasingly popular and powerful in Real World. This will allow the use of multiple sensors to monitor a variety of chronic and acute conditions simultaneously. In the current Situation, We have analysed, a huge amount of message are transmitted between layers. Data Acquisition node is used to collect the data from various nodes and sends data to the master node through wireless HART protocol which is also to minimize the data collision. Based on the priority given to the various parameters the master node will perform the task. Many nodes can be implemented to monitor and control the operation from the master node. The main advantage of this wirelessHART is the stack is implemented on a lowest two processor platform. In this method sensors are controlled by manual switches. Sensors values will be displayed in the display unit depending on the switching condition

VI. FUTURE WORK

In future, the experiment will be done in different OS like Linux, Neutrino and it can be implement in Android and Java Applications. In experimental Setup each sensors will be connected to the individual device.

REFERENCES

[1] P.M. Butala, Y.Zhang, R.C. Wagenaar, and T.D.C Little, "Wireless system for monitoring and real-time classification of functional activity", *in Proc. 2nd workshop on NetHealth 2012,IEEE Comsnets 2012,* Bangalore, India, Jan. 2012, PP. 1-5.

[2] P.Pawar, V.Jones, B.J.F Beijnum and H.Hermens, "A framework for the Comparison of mobile Patient monitoring system", *Journal of Biomedical Informatics*, Volume 45, Issue 3, June 2012, Pages 544-556.

[3] A. Cataldo, G. Cannazza, N. Giaquinto, A. Trotta and G.Andria, "Development of a remote system for real-time control of intravenous drip infusions", *in Proc. IEEE Int. Symp. MeMeA,* Bari, Italy, 2011, PP. 234-237.

[4] M.M Baig and H. Gholamhosseini, "Smart health Monitoring System: An Overview of Design and Modeling", *Journal of Medical Systems,* Vol. 37,no.2. PP. 1-14, 2013.

[5] Jianping Song; Song han; Mok, A.K;Deji chen; Lucas, M;Nixon, M A RTOS Based Reconfigurable Architecture for IWSN stack with Arm Cortex and EMF250RF Processor support. *International Journal of Science and Research*. 2014.

[6] Yi Yang, Ping Xiang, Mike Mantor, Huiyang zhou. CPU- assited GPU on fused CPU-GPU architectures, High Performance Computer Architecture (HPCA), 2012 *IEEE18th International Symposium,* 25-29 Feb, 2012.

[7] Liu Zhongyuan , Cui Lili, Ding Hong, Published a paper titled " Design of Monitors Based on ARM7 and Micro C/OS-II",2010 IEEE.

[8] Utz Roedig, Sarah Rutlidge, James Brown, Andrew Scott, "Towards Multiprocessor Sensor Nodes", *ACM Workshop on Hot Topics in Embedded Networked Sensors(HotEmNets'10),* 2010.

[9] Vehbi C. Gungor; Gerhard P. Hancke, Industrial Wireless Sensor Networks:Challenges, Design Principles, and Technical Approaches. *IEEE Transacation on Industial Electronics,* 2009,4258-4265.

[10] Zhang Yu, "Research on High Level Model and Performance Estimation", *Southeast University PHD thesis*, 2007.

